Learning over Molecules: Representations and Kernels

Learning over Molecules: Representations and Kernels

Jimmy Sun

A thesis submitted in partial fulfillment
for the degree of Bachelor of Arts in
Computer Science, Chemistry & Physics
Harvard College
Cambridge, MA

April 1, 2014

Abstract

In this paper, we tackle machine learning over molecular space by considering three representations for molecules: (1) a vector of molecular properties that we treat as predictor variables, (2) a graph that captures the relationship between individual atoms in a molecule, and (3) a cheminformatic fingerprint that "identifies" a molecule. We assess the viability of each representation by training a model to predict energy values. In particular, we look a class of models that use kernel methods, whereby the prediction algorithm relies on a similarity measure between training data. On a subset of the Harvard Clean Energy Project (CEP) database, we find a simple fingerprint similarity kernel to be the fastest and most accurate for predicting HOMO-LUMO energy gap values.

1 Introduction

Predicting molecular properties has been a long studied problem, particularly in the field of pharmaceutics and drug discovery. Recently, this interest has expanded to materials science in the search for next generation solar cells. Specifically of interest are organic, carbon based photovoltaic materials, which may be easier and cheaper to manufacture. Unfortunately, current carbon based photocells top out at around 5% percent energy conversion, which is too low for widespread use [1].

The research and development process for solar cells is expensive and time consuming, so a computational screening process for candidate materials is desirable. Current state-of-the-art quantum calculations take days to compute energetic properties for a single molecule, which is too slow for high throughput screening. Recently, the Harvard Clean Energy Project has taken the initiative to crowd-source such computations, constructing a database of over 2 million molecules and their calculated energies [2]. Given this repository of examples, we would like to leverage machine learning tools to build fast, accurate predictors for these properties of interest.

Machine learning over molecules is a unique and challenging problem due to the inherent nature of the molecular space. Unlike in many domains, here a clear physical process, the Schrodinger equation, governs the system. While the exact equation is difficult and computationally expensive to solve, the fact that an underlying model exists is appealing for machine learning. On the other hand, this problem domain is difficult from a technical point of view. Most standard regression techniques model a target variable as a function of some set of input predictor variables, but it is not immediately obvious how to do so when the input is a molecule. The key question, then, is how to best represent molecules for machine learning problems.

In this paper, we address this problem by considering three representations: (1) a vector of molecular properties that we treat as predictor variables, (2) a graph that captures the relationship between individual atoms in a molecule, and (3) a cheminformatic fingerprint that "identifies" a molecule. We assess the viability of each representation by training a model to predict energy values. In particular, we look a class of models that use kernel methods, whereby the prediction algorithm relies on a similarity measure between training data. This similarity metric, called a

kernel function, allows us to compare various representations using the same model by simply specifying a function that determines how similar two objects are under a given representation.

2 Related Work

2.1 Quantitative structure-activity relationship

In the fields of computational and medicinal chemistry, molecular properties are modeled under the quantitative structure-activity relationship (QSAR) framework [3]. Historically, properties of interest, such as biological activity of candidate molecules, have been modeled as a function of molecular properties. In particular, studies have focused on the pharmaceutical efficacy or toxicity of candidate molecules, often in a binary classification manner [4]. These studies have used both parametric models as well as nonparametric kernel methods, but often suffer from small sample sizes and noisy, empirical data.

2.2 Neural network predictions

Recently, Montavon et al. [5] have shown Coulomb matrices to be a useful representation for energetic predictions using neural networks. In particular, they propose the idea of random sampling of Coulomb matrices over the possible permutations of atomic indexing. Their best neural network predicted atomization energies significantly better than various kernel methods, but neural networks are difficult and time-consuming to train. Furthermore, they show that kernel methods are less affected by the specific representation of the Coulomb matrix (eigenspectrum vs. sorted vs. randomized, see section 3.2). We borrow their notion of a Coulomb matrix and analyze two possible kernels over them.

2.3 Graph kernels

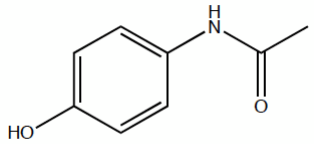
Graph similarity is an active branch of graph theory. Due to the large state space and relational nature of graphs, efficient computation of graph kernels is an important issue. One appealing similarity measure is the idea of graph edit distance, which

is the number of edit operations required to make one graph into another. This problem is unfortunately NP-hard, though upper and lower bounds can be computed in polynomial time [6]. Another class of graph kernels involve substructure similarity, which relates closely to a molecular fingerprinting method we will use [7].

3 Methods

3.1 Data

We use a subset of data provided by The Harvard Clean Energy Project, an initiative at Harvard University to identify organic molecules with promising photovoltaic properties. The entire data set features over 2 million molecules with energetic properties calculated through crowd-sourced quantum computations. These molecules are given in string representation called the Simplified Molecular-Input Line-Entry System (SMILES). This is one of the industry standards for molecular representation, and the initial input we must work with(Fig. 1). The response variable we attempt to predict is the difference in energy between the highest occupied molecular orbital (HOMO) and lowest unoccupied molecular orbital (LUMO). This HOMO-LUMO gap energy can be used as a proxy for the photovoltaic efficacy of a molecule [8].



c1c(O)ccc(NC(=O)C)c1

Figure 1: Example molecule and associated SMILES string.

3.2 Representations

Working with molecular SMILES directly is difficult since regression is usually tailored to a vector of predictor variables, rather than strings. We consider three representations more amenable to machine learning methods.

Cheminformatic features

Since each SMILES string represents a physical molecular compound, one approach to molecular representation is via chemical-physical properties of the underlying molecule. Such features may be simple descriptors such as counts of the number of carbon atoms, or more complex properties such as the pH. Using the cheminformatic tool suite ChemAxon [12], we can extract a variety of real-valued properties directly from SMILES strings. We use all 34 available composite molecule properties (we ignore properties that are given as per-atom), shown below[1].

Table 1: 34 Extracted ChemAxon features

Mass	Molecular polarizability	axxPol
ayyPol	azzPol	ASA
ASA+	ASA-	ASA_H
ASA_P	Dreiding energy	fsp3
Harary index	Hyper wiener index	Max projection area
Max projection radius	Length perpendicular to max area	Min projection area
Min projection radius	Length perpendicular to min area	MMFF94 energy
Platt index	Van der Waals surface area (2D)	Polar surface area
Randic index	Van der Waals surface area (3D)	Szeged index
Wiener polarity	Octanol/water partition coefficient	Acceptor count
Donor count	Acceptor site count	Donor site count
Atom count		

It is important to note that some of these features are in fact energetic calculations. However, these calculations appear to be fast approximations (per-molecule extraction time is on the order of minutes for all 34 features), so using these features is still viable for predicting our target energies.

Molecular Graphs

We can also treat a molecule as a graph, using individual atoms as the nodes. In an adjacency matrix representation, edges represent bonds within the molecule, with the edge value indicating the type of bond, e.g. 1 – single bond, 2 – double bond, 1.5 – aromatic bond. This method captures important bonding interactions in the

[1]Descriptions can be found at `http://www.chemaxon.com/marvin/help/chemicalterms/EvaluatorFunctions.html`.

molecule, but loses information since those chemical bonds are simplified constructs in and of themselves.

Another representation that captures the actual geometry of a molecule is a Coulomb matrix. Individual atoms are still treated as nodes, and edges weights are given by the energetic interactions between pairs of nodes:

$$C_{ij} = \begin{cases} 0.5Z_i^{2.4} & i = j \\ \frac{Z_i Z_j}{|R_i - R_j|} & i \neq j \end{cases} \tag{1}$$

where Z_i is the nuclear charge of atom i, and R_i is its Cartesian coordinate in space. The geometry of an atom can be quickly retrieved from SMILES using energy-minimization techniques via a cheminformatic Python library called RDKit [9].

In reality, the interactions between atoms is much more complex and more accurately falls somewhere between these two representations. However, to first order, these graphs provide a suitable approximation to the topology of a molecule.

One important issue with working with molecular graphs is the labeling of nodes. A single molecule can produce many matrices based on the indexing of individual atoms, and it is not immediately obvious how to ensure consistency across molecules.

A simple approach is to let further reduce the representation by using the sorted eigenvalues of an adjacency/Coulomb matrix. The eigenspectrum is invariant to row/column permutations so it will not depend on indexing choices. The downside of this method is that we are potentially throwing out too much information to recover accurate predictions. To use the actual Coulomb matrix, we can propose an indexing such that the sum of each row is greater than or equal to all subsequent rows. This can be computed quickly and provides some semblance of comparability between individual elements of the matrix.

Finally, we need to deal with the fact that Coulomb matrices will vary in size based on the number of atoms in a molecule. For kernels that require matching dimensionality, we can add "dummy atoms" with a nuclear charge of 0 to smaller molecules. This essentially pads 0s onto the adjacency/Coulomb matrices up to the largest molecule in the data set.

Molecular fingerprints

The last representation we use is a fingerprinting method that accounts for substructures in molecules. Here we use a path-based fingerprint implemented in OpenBabel [10]. This method finds all atomic chains up to length 7 in a molecule, accounting for bond type, order, and cycles (e.g. N=C–C is equivalent to C–C=N but not N–C–C or N–C=C). Each canonical fragment is hashed to set 1024 bit vector (Fig. 2). Thus a molecular fingerprint indicates the presence or absence of substructures within a molecule. In addition to fixed-size fragments, user-specified substructures can be used, allowing for input of prior knowledge. Such fingerprinting methods have been widely used for comparing molecules in medicinal chemistry [11].

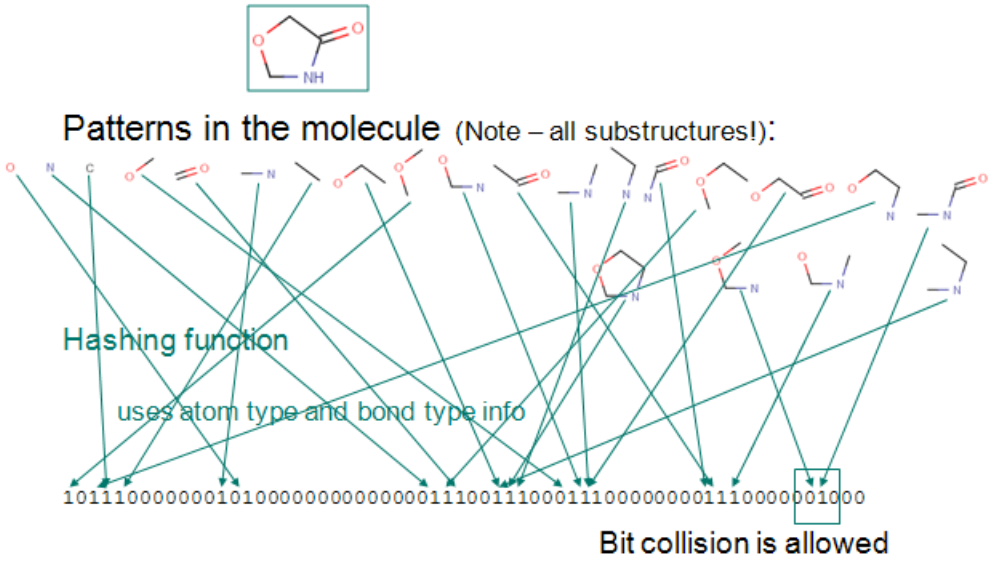


Figure 2: Molecular fingerprinting [12].

3.3 Gaussian process model

Given the inherently nonlinear interactions governing molecular systems, we use a Gaussian process for regression. At a high level, a Gaussian process acts as a nonlinear interpolater to data, modeling some smooth underlying function [13]. Here we give a brief mathematical overview.

Suppose we want to infer function $f : x \to \mathbb{R}$, e.g. f would map a molecule to an energy value. A Gaussian process assumes that the realization of f at a finite but arbitrary number of points is jointly Gaussian, i.e.

$$(f(x_1), f(x_2), \ldots, f(x_N)) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{2}$$

where the covariance $\Sigma_{ij} = K(x_i, x_j)$ is a kernel function. Essentially, if x_i, x_j are similar with respect to the kernel, then we would expect $f(x_i), f(x_j)$ to be similar. Fitting a Gaussian process involves inverting the kernel matrix, and is thus $O(N^3)$ in the training set size.

Note that we have never specified a form for x; all we require is a valid (positive semi-definite) kernel that measures similarity between two x's. This is appealing since we can predict energy as a function of a molecule directly, without dealing with its specific representation. Given this model, the most important consideration is the choice of a kernel. Different representations lend themselves naturally to different kernels, and here we consider three kernels.

RBF kernel over feature vectors

Given two feature vectors $\boldsymbol{x_1}, \boldsymbol{x_2}$, a standard kernel is the radial basis function (RBF) kernel, given by

$$K(\boldsymbol{x_1}, \boldsymbol{x_2}) = \exp(-\frac{||\boldsymbol{x_1} - \boldsymbol{x_2}||_2^2}{2\sigma^2}) \tag{3}$$

This kernel functions as a similarity measure between the feature vectors since it is a function of the Euclidean distance between them. The hyper-parameter $l = \frac{1}{2\sigma^2}$ controls the scale of "closeness" between feature vectors, and can be tuned via cross-validation. The RBF kernel is a natural choice for the cheminformatic feature representation, but can also be used for the sorted Coulomb matrix by flattening the matrix into a sorted vector.

Graph kernels

Given the physical interpretation of Coulomb matrices as a molecular graph, a more sophisticated approach might be to come up with a measure of similarity over graphs.

Here we consider an approach called the random walk graph kernel [14]. The idea is simple: given two graphs, perform random walks on both, and count the number of matching walks. This yields a somewhat intuitive measure of similarity between two graphs. One appealing interpretation of this random walk is of an electron "diffusing" randomly around the molecule, where the random walk tendencies might be analogous to an electron obeying the wave equation. First we introduce some concepts and notation.

Direct product graphs. Performing simultaneous random walks on two graphs is equivalent to performing one random walk on the direct product graph $G_\times = G_1 \otimes G_2$, where

$$V_\times = \{(v_i, v'_r) : v_i \in V_1, v'_r \in V_2\} \tag{4}$$

$$E_\times = \{((v_i, v'_r), (v_j, v'_s)) : (v_i, v_j) \in E_1 \wedge (v'_r, v'_s) \in E_2\} \tag{5}$$

For our unlabeled molecular graphs, the weight matrix is the Kronecker product of the two individual weight matrices $W_\times = W_1 \otimes W_2$.

Starting and stopping probabilities. To perform such random walks, valid starting and stopping distributions p, q must be assigned over the graph. For our purposes, we stick with uniform distributions over each molecule. Then we can let $p_\times = p_1 \otimes p_2, q_\times = q_1 \otimes q_2$ be the starting and stopping distributions over the direct product graph that we will work with.

Kernel definition. From the direct product graph of dimension n', the $(i-1)n'+r, (j-1)n'+s)$ entry of $W_\times^k$ represents the probability of simultaneous length k random walks from v_j to v_i on G_1 and from v'_s to v'_r on G_2. Then we can formally define a random walk kernel on G_1, G_2 to be

$$K(G_1, G_2) = \sum_{k=0}^{\infty} \mu(k) q_\times^T W_\times^k p_\times \tag{6}$$

where $\mu(k)$ is a coefficient that weights random walks of length k. Here we use a geometric coefficient $\mu(k) = \lambda^k$, which ensures that the above sum converges for proper choice of μ. This guarantees a positive semi-definite Gram matrix which is necessary for kernel methods.

In addition, using the fact that our molecular graphs are unlabeled, we can make

use of a spectral decomposition method to efficiently compute this sum:

$$K(G_1, G_2) = \sum_{k=0}^{\infty} \mu(k) q_\times^T (P_\times D_\times P_\times^{-1})^k p_\times = q_\times^T P_\times (\sum_{k=0}^{\infty} \mu(k) D_\times^k) P_\times^{-1} p_\times \tag{7}$$

where $W_\times = P_\times D_\times P_\times^{-1}$. Since $D_\times$ is a diagonal matrix, we can quickly compute the infinite sum, e.g. for geometric μ:

$$K(G_1, G_2) = q_\times^T P_\times (\boldsymbol{I} - \lambda D_\times)^{-1} P_\times^{-1} p_\times \tag{8}$$

where inverting the diagonal matrix is trivial. Such a calculation would take $O(n^6)$ time per kernel element, due to the cubic complexity of matrix inversion, and the n^2 size of the direct product graph. However, this can be sped up to $O(n^3)$ per kernel element by pre-decomposing the individual matrices:

$$K(G_1, G_2) = (q_1^T P_1 \otimes q_2^T P_2)(\sum_{k=0}^{\infty} \mu(k)(D_1 \otimes D_2)^k)(P_1^{-1} p_1 \otimes P_2^{-1} p_2) \tag{9}$$

Finally, we must deal with choosing the parameter λ. It must be small enough for the sum to converge, but not so small as to completely discount longer walks. This turns out to be quite difficult for the molecular graph matrices we are working with, as we will see in the results.

Fingerprint similarity

The last kernel we use is the Tanimoto similarity metric. This is a straightforward and natural similarity measurement for molecular fingerprints, defined as

$$K(f_1, f_2) = \frac{N_{12}}{N_1 + N_1 - N_{12}} \tag{10}$$

where N_1, N_2 are the number of bits set in f_1, f_2 respectively, and N_{12} are the number of set bits common to both. This kernel has the advantage of already being in the range $[0, 1]$, and does not require additional parameter tuning.

3.4 Experimental setup

We use a subset of 1000 molecules from the Clean Energy Project data set, split into a 800 for training and 200 for testing. Coulomb matrices and cheminformatic features are pre-processed so kernels can be calculated from their respective representations directly. Hyper-parameters are selected via cross-validation, and mean-squared errors are recorded.

4 Results

First we present the test accuracy of the representations and kernels used (Table 2) as well as a mean predictor reference. Despite the richness of the Coulomb matrix representation, it performs poorly on this subset of data. Molecular fingerprinting results in the smallest error, while random walk kernels are the worst.

Representation	**Kernel**	**Mean squared error**
mean predictor	–	0.089
cheminformatic features	RBF	0.022
adjacency eigenvalues	RBF	0.019
sorted adjacency matrix	RBF	0.023
Coulomb eigenvalues	RBF	0.054
sorted Coulomb matrix	RBF	0.039
adjacency matrix	random walk	0.051
Coulomb matrix	random walk	0.065
molecular fingerprint	Tanimoto coefficient	0.015

Table 2: Predictive accuracy of various representations and kernels

In addition, kernel matrix computation times are shown below. RBF kernels over feature vectors and molecule fingerprints can be calculated very quickly, but random walk kernels are much slower.

	RBF kernel	Random walk kernel	Fingerprint similarity
Computation time	10–15 s	2 hr	12 s

Table 3: Time to compute 800×800 kernel matrix on training data.

5 Discussion

The convergence of random walk sums appear to be responsible for the poor performance of random walk kernels. To ensure that the sums converge, we use λ on the order of 1×10^{-8} for Coulomb matrices and 1×10^{-2} for adjacency matrices. However, such a small λ decreases the weight of longer walks, causing kernel values to not vary enough across different pairs of graphs. Fig. 3 shows a scatter of kernel values against target energy differences. The distribution of kernel values for the random walk kernel is very narrowly clumped around 1, and shows no correlation between kernel values and target variable differences; fingerprint kernel values, on the other hand, are well distributed from 0 to 1, and the correlation between kernel value and target distance is negative as we would hope (i.e. $K(x_i, x_j) \approx 1$ would imply that $|f(x_i) - f(x_j)| \approx 0$). Smaller values of λ cause even more tightly clumped kernels, but increasing λ eventually yields a divergent sum that results in singular kernel matrices. Given both the poor performance and slow computation time, random walk kernels as we have formulated them seem to be a poor choice for this learning problem.

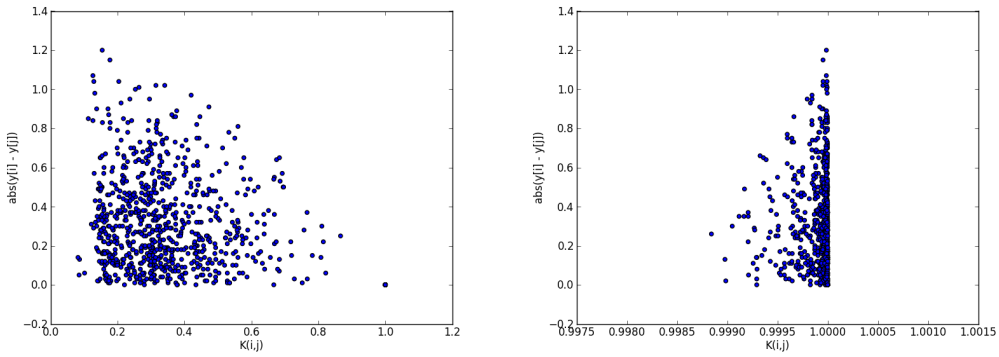


Figure 3: Kernel value vs. energy difference for fingerprint similarity (left) and Coulomb random walk (right).

Though Coulomb matrices performed poorly on this small data set, their utility in [5] suggest that as a representation, they are still worth studying. From Fig. 4, it appears that increasing the training set size may yield better performance for RBF kernels over both matrix representations, so graph based methods representations

may be relatively better at larger scale. It may also be that an RBF kernel is not ideal for sorted matrices where some rows/columns might be artificially padded 0s.

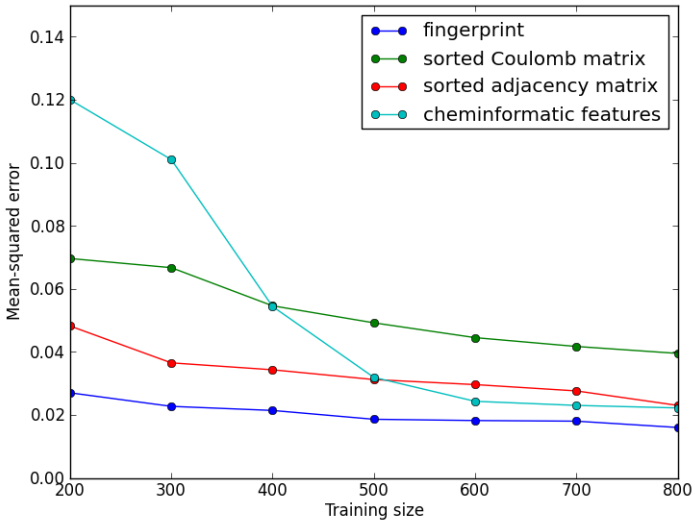


Figure 4: Performance vs. training set size.

It is interesting that the fingerprint kernel had the smallest error on our data set. On one hand, this somewhat unsatisfying due to the artificial hash construction of such fingerprints. On the other hand, it does highlight the importance of substructures in a molecule, which makes physical sense: chemical fragments such as rings or functional groups often impart specific properties onto molecules, so their impact on HOMO-LUMO energies is not unreasonable. Furthermore, this suggests that we could consider tuning the fingerprinting representation by choosing specific substructures of importance. Identifying useful substructures and tuning the fingerprinting process is a promising direction for future work.

6 Conclusions

In this paper we have examined representations and kernels for machine learning over molecules. We considered cheminformatic feature vectors, graph based matrix representations, and molecular fingerprints. Along with these representations, we considered a simple RBF kernel, a random walk graph kernel, and a fingerprint similarity index. On a subset of CEP data, molecular fingerprinting predicted HOMO-LUMO gaps with the lowest error. The fast computation and flexibility of this representation/kernel is promising, and further study is warranted. Though the random walk kernel formulation we used did not prove successful, graph-based representations such as the Coulomb matrix may still be of interest. The primary goal of this study was exploratory, and we have established a foundation for further work.

References

[1] Martin A. Green, Keith Emery, Yoshihiro Hishikawa, Wilhelm Warta, and Ewan D. Dunlop. Solar cell efficiency tables (version 42). *Progress in Photovoltaics: Research and Applications*, 21(5):827–837, 2013.

[2] Johannes Hachmann, Roberto Olivares-Amaya, Adrian Jinich, Anthony L. Appleton, Martin A. Blood-Forsythe, Laszlo R. Seress, Carolina Roman-Salgado, Kai Trepte, Sule Atahan-Evrenk, Suleyman Er, Supriya Shrestha, Rajib Mondal, Anatoliy Sokolov, Zhenan Bao, and Alan Aspuru-Guzik. Lead candidates for high-performance organic photovoltaics from high-throughput quantum chemistry - the harvard clean energy project. *Energy Environ. Sci.*, pages –, 2014.

[3] Chanin Nantasenamat, Chartchalerm Isarankura-Na-Ayudhya, Thanakorn Naenna, and Virapong Prachayasittikul. A practical overview of quantitative structure-activity relationship. *EXCLI*, 8:74–88, 2009.

[4] S. Joshua Swamidass, Johnathan Chen, Jocelyne Bruand, Peter Phung, Liva Ralaivola, and Pierre Baldi. Kernels for small molecules and the prediction of mutagenicity, toxicity, and anti-cancer activity. *Bioinformatics*, 21:359–368, 2005.

[5] Grgoire Montavon, Katja Hansen, Siamac Fazli, Matthias Rupp, Franziska Biegler, Andreas Ziehe, Alexandre Tkatchenko, Anatole von Lilienfeld, and Klaus-Robert Mller. Learning invariant representations of molecules for atomization energy prediction. In Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Lon Bottou, and Kilian Q. Weinberger, editors, *NIPS*, pages 449–457, 2012.

[6] Zhiping Zeng, Anthony Tung, Yianyong Wang, Jianhua Feng, and Lizhu Zhou. Comparing stars: On approximating graph edit distance. *VLDB*, 2009.

[7] Graph kernels for chemical informatics. *Elvesier*, 2005.

[8] Jonathan D. Servaites, Mark A. Ratner, and Tobin J. Marks. Practical efficiency limits in organic photovoltaic cells: Functional dependence of fill factor and external quantum efficiency. *Applied Physics Letters*, 95(16):–, 2009.

[9] Greg Landrum. Rdkit: A software suite for cheminformatics, computational chemistry, and predictive modeling. `http://www.rdkit.org/`, 2013.

[10] Open babel: The open source chemistry toolbox. `http://openbabel.org/wiki/Main_Page`, 2013.

[11] Gerald Maggiora, Martin Vogt, Dagmar Stumpfe, and Jurgen Bajorath. Molecular similarity in medicinal chemistry. *Journal of Medicinal Chemistry*, 2013.

[12] ChemAxon Ltd. Chemaxon cheminformatics toolchain suite. `https://www.chemaxon.com/`, 1998–2013.

[13] Albert P. Bartók, Mike C. Payne, Risi Kondor, and Gábor Csányi. Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons. *Phys. Rev. Lett.*, 104:136403, Apr 2010.

[14] Karsten Borgwardt, Risi Kondor, Nicol Schraudolph, and SVN Vishwanathan. Graph kernels. *Journal of Machine Learning Research*, 11:1201–1242, 2010.